

Install dockers and execute docker commands to manage images and interact with containerization.

step-1) Installing docker +

- 1). Search docker ~~Tools or~~ download in google
- 2) Go to the github repo of docker-archives, to download latest version of docker
- 3) click and install the exe file in the repo. ~~to~~
- 4). Go to the above downloaded files and click on next button to uncheck download. uncheck virtualbox if already it is present in ur system.
- 5) Now click on the "Docker quickstart Terminal", It will run to check pre-create checks. and it create a machine
- 6) Now, we can see the bash in the Docker terminal

step-2) Execution +

- >> docker run hello-world // runs the hello-world files in the library.
- >> docker ps // we can see what are the files running
- >> docker ps -all // To see all the processes including excited

Teacher's Signature : _____

>> docker stop <container id> // To stop the process.

>> docker kill <container id> // To kill the process (acts like force stop)

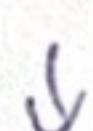
* docker run = docker ^{create} start + docker stop

\$ docker run redis



\$ docker ~~run~~ create redis

>> "Token id"



\$ docker start "token id"

\$ docker start -a // To run

\$ docker exec -i -t "docker-id" sh // To open the shell

ls // we can see files in redis. in any running file

** click "ctrl+d" to exit from docker shell.

To learn docker file instruction to build an image for sample web application using docker file

To create an image

Step-1 choose a base image

Step-2 Run command for your software

Step-3 specify the startup command.

Step-4

1) create a file and open it in VS code

2) Go to terminal in VS code, type `>>pwd` to get path of the VS code

3) copy the path and open docker terminal to execute,

\$

4) create a "Dockerfile" in the VS code, and type the below lines.

From alpine //choosing base image

Run apk add --update redis // download & install dependencies

CMD ["redis-server"] // setup the startup command

5) Now open Terminal in VS code and execute
~~\$~~ This command, to run all the steps in "Dockerfile"
`>> docker build`.

Create a jenkins CI/CD pipeline with SonarQube integration to perform static code analysis.

Step-1 & Installing jenkins

- 1) Search for jenkins in browser & download jenkins on windows
- 2) Click on the .msi file which has been downloaded and clicking few general next will be good enough.
- 3). To open jenkins on your web browser, click on the port link.
- 4) Go to the "(:)\\programData\\Jenkins\\secrets\\initialAdminPassword" to view password, copy the password, paste it and get logged in.
- 5). Now click on "install suggested plugins" to customize jenkins.
- 6). Create your account by providing basic details.
- 7) By clicking on start using jenkins, we can easily use jenkins.

Step-2: Integration of sonarcube with jenkins

- 1) Generating token on sonarcube, for that we need to create a project & generate a token.
- 2) Saving the token in jenkins, for that go to manage jenkins, then credentials → click on new credential → provide the credentials by selecting kind as "secret text"
- 3) Now, the token will be created in jenkins
- 4) Go to dashboard → manage jenkins → system → check the box in SonarQube Servers
- 5) For Server URL, provide the url with IP in this way "http://IP address:port number", once we add the details save it to add server details of Sonarcube
- 6) Now, to download plugin, go to Available plugins and search for sonarcube and install, using this we can perform anything from sonarcube & the integration is completed.

Step-3) Trigger Sonarcube analysis from jenkins.

- 1) creating pipeline, from jenkins dashboard click on new item and enter the name, select free style project and click on "ok" to create it.
- 2) In General, source code management, select git and give the repository URL, change branch to "main", under build environment click on "Sonarcube Scanner environment" and select the token which we created and click on save under Analysis properties we have to provide the following details :

```
{
  sonar.projectKey = retail"key"
  sonar.projectName = " "
  sonar.projectVersion = "1.0"
  sonar.language = java
  sonar.test = src/test/java
  sonar.sources = src/main/java
}
```

Then click on Save

- 3) Now, just click on build and from console output we can see logs

- 4) Now, in Sonarcube, we can see the report being created

- 5) By clicking on code Smells, we can see abt the issues and in the home page we can see the report.

Teacher's Signature : _____

Build a pipeline of jobs using Maven in Jenkins.

Step-1: Install Jenkins pipeline:

1. Go to Jenkins dashboard.

2. Navigate to Manage Jenkins > Manage Plugins

3. Install the following plugins:

- Pipeline
- Maven Integration

Step-2: Configure Maven in Jenkins:

1. Navigate to Manage Jenkins > Global Tool Configuration.

2. Under Maven, click Add Maven

- provide a name (eg: "Maven 3.8")
- specify the Maven installation path or let Jenkins install it automatically.

Step-3: Create a new Jenkins pipeline Job

1. Go to New item

2. Enter a name (eg: "maven-pipeline")

3. Select Pipeline and click OK.

Teacher's Signature: _____

Config

Step-4 Configure git repository

1. In the pipeline configuration page, under pipeline definition, choose:

- pipeline script from SCM if using a Jenkins file.
- Pipeline script for manual script entry

2. Add the git repository URL containing the java project and pom.xml.

Step-5 writing pipeline script

```

pipeline {
    agent any
    environment {
        PATH = "/opt/apache-maven-3.6.3/bin:$PATH"
    }
    stages {
        stage ("clone code") {
            steps {
                git credentialsId: 'git-credentials',
                url: 'https://github.com/username/project-repo.git'
            }
        }
        stage ("build code") {
            steps {
                sh "mvn clean install"
            }
        }
    }
}

```

Teacher's Signature : _____

step-6 to save and run the pipeline

1. Save the configuration
2. Click Build Now
3. Monitor the console output for job status.